N90-25534

Process and Information Integration via Hypermedia

David G. Hammen
Daniel L. LaBasse
Robert M. Myers

The MITRE Corporation, Houston, Texas

Abstract

Success stories for advanced automation prototypes abound in the literature but the deployments of practical large systems are few in number. There are several factors that militate against the maturation of such prototypes into products. This paper addresses one issue, namely, the integration of advanced automation software into large systems.

Advanced automation systems tend to be specific applications that need to be integrated and aggregated into larger systems. Systems integration can be achieved by providing expert "user-developers" with verified tools to efficiently create small systems that interface to large systems through standard interfaces.

This paper explores the use of hypermedia as such a tool in the context of the ground control centers that support Shuttle and space station operations. Hypermedia can be an integrating platform for data, conventional software, and advanced automation software in hypermedia enables data integration through the display of diverse types of information and through the creation of associative links between chunks of information. Further, hypermedia enables process integration through graphical invoking of system functions. Through analysis and examples we are able to illustrate how diverse information and processing paradigms can be integrated into a single software platform.

Integrating Small and Large Systems

Small Systems Integration

Success stories for advanced automation prototypes abound in the literature but the deployments of practical large systems are few in number. There are several factors that militate against the maturation of such prototypes into products. One factor is that advanced automation systems tend to be specific, relatively small, applications that need to be integrated and aggregated into larger systems.

Application software that supports small groups of users or narrow disciplines is sometimes viewed as a threat to large systems. That is, severe controls imposed by system developers and maintainers of large systems (of necessity to ensure system integrity) are impediments to the development of small systems. By providing the user with verified tools and standard interfaces, systems integrators can enable the development of small systems that integrate well into large systems.

We perceive a need for a defined transition point between the systems integrators and the users that allows the users to automate their operations themselves. By providing a verified toolkit to expert users ("user-developers"), we can enable the user to streamline local operations without generating new requirements or change requests against the large system. That is, authority and responsibility for localized changes which affect small groups of users can be accomplished by small groups within the constraints of the toolkit and their management. This does not imply that configuration management (CM) should be abandoned for small systems. On the contrary, CM is a shared responsibility of the system developers and the users.

Systems Integration in Large Systems

A large group of people is needed to perform the systems integration task for a system as complex as a spacecraft ground control center. The users of small systems are usually not interested in, or qualified for, this large systems integration job. The systems integrators focus on widely shared system functions such as operating systems, data distribution, user

interface management, and configuration management. As a widely shared resource, an automation tool would be procured, verified, and tailored for control center use by the systems integrators. In addition to these verified automation tools, the user-developers must be provided with standard interfaces to the large system.

The large system into which advanced automation techniques will be incorporated for the Mission Control Center Upgrade (MCCU) can be best described as a "hardware independent software environment" or HISE. This is a run-time environment characterized and defined by industry standards. The standards that compose the HISE are the UNIX System V Interface Definition (SVID) for operating system services, X-Windows for the user interface, the Graphics Kernel System (GKS) for graphics, the Open Systems Interconnection (OSI) protocols for networking, and ANSI 'C' as the high level language. HISE manifests itself to the developer as libraries of function calls which provide services in an agreed-upon interface. The system is hardware independent in that the system need only provide the interfaces and implement the services as agreed upon in the standards. The HISE is a Level C requirement of the MCCU 2.5 delivery [JSC 1988].

The next section explores the use of hypermedia as one such automation tool that can be provided to user-developers.

Information Presentation via Hypermedia

Introduction

The term "Hypertext" describes a unified information system for non-sequentially accessing text and/or graphics. "Hypermedia" expands these concepts to include other media such as sound, photographs, and video. Hypermedia can be further extended. This paper investigates the possibility of using hypermedia as an integrating platform for executable processes as well as for information.

The use of hypermedia is well suited to needs of spacecraft ground control center where non-sequential access to large amounts of diverse information is the norm. In addition to their role as a sophisticated user community, flight controllers are also a sophisticated development community. Hypermedia meets well the development needs of a user community that wants to automate their operations themselves.

Visionary views of hypermedia typically involve far-reaching, ad-hoc searches of large collections of data by scholarly researchers [Bush, 1967; Nelson, 1987]. These visionary views are not particularly applicable to the more controlled environment of a manned space flight control center; they may even be seen as exemplifying an unacceptable technology.

Flight controllers use large amounts of diverse but linked information, but this use is more regimented than the scholarly access of data that is exemplified in these visionary views [Johns, 1987]. Hypermedia is as applicable to this more regimented use of data as it is to scholarly research. Any electronic form of flight documentation must display diverse kinds of information such as text, graphs, and schematics; many hypermedia systems excel at this. Each piece of the flight documentation is linked to other pieces (such as rules and rationales) in a variety of ways, and the handling of these linkages is the premier characteristic of all hypermedia systems.

An Example

Flight controllers direct crew operations according to predetermined plans. The crew procedures that compose the plan can be displayed electronically in various forms, for example as a timeline or as a time-tagged list. A hypermedia system could allow a controller to select an upcoming procedure directly from the plan and display the steps that need to be followed to carry out the procedure. The controller could also follow hypermedia links to examine associated procedure rationale, constraints, and supporting material such as layouts of onboard switch panels, diagrams depicting the Shuttle's attitude, or schematics of onboard systems.

Paper vs. Screen

Flight controllers currently use paper documents in Shuttle Mission Control Center (MCC). These documents are now being produced electronically, yet the final product of this process is still paper documents. Although the electronic medium is preferable for the production of the product it apparently is not for the actual use of the data. While tradition may be partially responsible for the preference for paper, another reason for this choice is that paper is often the better medium [Yankelovich, 1985; Johns, 1989].

Books inherently provide contextual information that is not conveyed well by most electronic information systems. For example, placemarkers may be easily attached to paper pages; notes may be easily written on paper pages [Yankelovich, 1985].

Hypermedia implementations of information systems can provide these advantages of the printed media while accentuating the advantages of the electronic media. Contextual information can be

provided with graphical cues. Annotations and markers can be added to a hypermedia display. [Bernstein, 1988]

Potential Problems

There are some problems that must be solved before hypermedia can be accepted for use in the control center. Controllers often do not have the luxury of making the wide-ranging searches that are proposed by hypermedia visionaries. The links must be carefully constructed to aid the controllers in performing their jobs [Johns, 1987]. This places an extra burden on the organizers of the documentation (beyond the primary task of collecting and designing the information). The data for existing paper products typically is not chunked to suit the needs of a hypermedia implementation nor are the links well enough defined, further extending the burden placed on the organizers of the documentation [Raymond, 1987].

Finally, a hypermedia implementation in an operational flight control environment must overcome another advantage of paper not mentioned above. Paper is static and easily controlled [Yankelovich, 1985]. A hypermedia system for the control center must have configuration control capabilities.

Process Integration via Hypermedia

We propose that the hypermedia also be used to integrate executable processes with information. There are many ways to encapsulate the lower level invocations of operating systems functions within a higher level construct. Hypermedia systems not only can accomplish this encapsulation but can also link these encapsulations with related documentation.

Reducing Information Overload

Hypermedia can provide a graphical front end to system services [Conklin, 1987]. This could reduce manual activities (such as typing the name of a program and associated data files) and thereby help to keep the user's mind on the task being performed. As with the information integration system, the process integration system should be built by the users; only they know the processes to be performed and the links between them and the supporting documentation.

An Example

A controller may need to execute a specialized program to analyze suspect components in response to sensed anomalies. One way to do this is to tell the operating system to execute the program by typing in the name of the program and associated data files. Another way is to select a graphical icon for the program. This latter technique helps keep the user focused on the analysis. A hypermedia approach extends this technique by associating the icon with other objects that are related to the suspect components. The controller could transfer from the static hypertext documentation to the active analysis program by selecting the appropriate item from the hypermedia display that integrates the documentation with the process icon.

How to Proceed

The controllers' flight procedures and related material are excellent candidates for a hypermedia information access system. It is important to get the users involved in development and evaluation of such a system. Only the users know what the data is, how to "chunk" the data into meaningful information, and what links to establish between the chunks of information. The users of this system could also be the small systems builders, user-developers, performing the systems integration task using hypermedia as a systems integration toolkit. The systems integrators of the large systems need to support this activity by acquiring the hypermedia tool, verifying it for space-rated use in a control center, and providing it to the user-developers.

Summary

We perceive a need for a defined hand-off point between the systems integrators and the users that allows the users to automate their operations themselves. This can be achieved by providing user-developers with verified tools, such as hypermedia, by which they can efficiently create small systems that interface to large systems through standard interfaces.

Acknowledgements

We thank Ken Jenks and Mark Dean of Rockwell Shuttle Operations Co. for their suggestions on the applicability of hypertext to the flight data file. We thank Dona M. Erb of the MITRE Corporation who provided background information on hypermedia and critiqued this paper.

This project is a part of NASA Research, Technology, Objectives and Plans (RTOP) task 488-50 and is administered by the Systems Development Division at the NASA Johnson Space Center.

References

Bernstein, M. (1988), *The Bookmark and the Compass: Orientation Tools for Hypertext Users*, ACM SIGOIS Bulletin, Vol. 9, No. 4, October 1988, pp. 34-45

Bush, V. (1967), *Memex Revisited*, Science Is Not Enough, V. Bush, ed., New York, NY: William Morrow & Co., Inc., Apollo Edition, 1969

Conklin, J. (1987), *A Survey of Hypertext*, MCC Technical Report STP-356-86, Rev. 2, Austin TX: Microelectronics and Computer Technology Corporation, December 1987

Johns, G. (1987), *Flight Data File for the Space Station, Volume I: Baseline Definition*, MTR-10019, Vol 1, Houston TX: The MITRE Corporation, February 1987

Johns, G. (1989), *Dynamic Display of Electronic Crew Procedures for Space Station*, Journal of Spacecraft and Rockets, Vol. 26, No. 1, January-February 1989, pp. 39-46

JSC (1988). *Workstation Executive (WEX) Requirements for Delivery 2.5*. JSC-22363. October, 1988

Nelson, T. (1987), *Literary Machines*, Abridged Electronic Edition 87.1, Macintosh hypertext edition in GUIDE, Bellevue WA: OWL, International, Inc., 1987

Raymond, D.; Tompa, F. (1987), *Hypertext and the New Oxford English Dictionary*, HyperTEXT'87 Papers, Chapel Hill, NC: The University of North Carolina, 1987, pp. 143-153

Yankelovich, N.; Meyrowitz, N.; van Dam, A. (1985), *Reading and Writing the Electronic Book*, Computer, Vol. 18, No. 10, October 1985, pp. 15-30